UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/682,387	10/09/2003	Basu Vaidyanathan	AUS920030439US1	8372

40412 7590 07/21/2009

IBM CORPORATION- AUSTIN (JVL)
C/O VAN LEEUWEN & VAN LEEUWEN
PO BOX 90609
AUSTIN, TX 78709-0609

EXAMINER	
SYED, FARHAN M	

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
07/21/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte BASU VAIDYANATHAN and LARRY BERT BRENNER

Appeal 2008-005964
Application 10/682,387[1]
Technology Center 2100

Decided: July 21, 2009[2]

Before JAMES D. THOMAS, JEAN R. HOMERE, and JOHN A. JEFFERY, *Administrative Patent Judges.*

HOMERE, *Administrative Patent Judge.*

DECISION ON APPEAL

---

[1] The application was filed on October 9, 2003. The real party in interest is International Business Machine Corporation.

[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1 through 20. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

Appellants' Invention

Appellants invented a system, apparatus, and method for enhancing priority boosting of scheduled threads. (Spec. 4, ll. 3-4.) As depicted in the flowchart of Figure 2, the process starts when a second thread is dispatched to a second central processing unit ("CPU") for execution (200 and 202). (Spec. 8, ll. 10-13.) During execution, if the second thread realizes that it needs to place a lock on a shared resource being held by a first thread (204), the second thread checks to see if it has a higher priority than the first thread (208). (Spec. 8, 13-17.) If the second thread has a higher priority than the first thread, the second thread may boost the priority of the first thread by passing the first thread its own priority (210). (Spec. 8, ll. 17-19.) After the second thread passes its priority, the second thread checks to see whether the first thread is scheduled to run on a first CPU or a second CPU (212). (Spec. 8, ll. 20-23.) If the first thread is scheduled to run on the first CPU, the first thread may be rescheduled to be executed on the second CPU (214). (Spec. 8, ll. 23-25.) Thus, the objective of Appellants' claimed invention is to enhance priority boosting for a process that has a lock on a shared resource by obtaining CPU time for the first thread as soon as possible. (Spec. 3, ll. 16-19.)

Illustrative Claim

Independent claim 1 further illustrates the invention as follows:

> 1. A method of enhancing priority boosting of scheduled threads comprising the steps of:
>
> determining by a second thread being executed on a second CPU whether to wait for a lock on a shared resource held by a first thread that is scheduled to be executed by a first CPU, the second thread having a higher priority than the first thread;
>
> boosting the priority of the first thread by passing the higher priority of the second thread to the first thread if the second thread has to wait for the lock on the shared thread;[3] and
>
> enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU.

Prior Art Relied Upon

The Examiner relies on the following prior art as evidence of unpatenability:

Bak	US 6,167,424	Dec. 26, 2000
Gosalia	US 2004/0160446 A1	Aug. 19, 2004 (filed Jan. 22, 2004)

Rejections on Appeal

The Examiner rejects the claims on appeal as follows:

Claims 1 through 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Bak and Gosalia.

Appellants' Contentions

1. Appellants contend that the Examiner erred in concluding that the combination of Bak and Gosalia renders the claimed invention

---

[3] Based on our reading of the original disclosure, it is apparent that the recitation of the shared thread" in claim 1 is a typographical error. We therefore presume that the reference to "the shared thread" was intended to be directed to "the shared resource" as indicated in line 4 of claim 1.

unpatentable. In particular, Appellants argue that the subject matter relied upon in Gosalia is not supported by U.S. Provisional Patent Application 60/448,402. (App. Br. 5.)

2. Appellants contend that the Examiner erred in concluding that the combination of Bak and Gosalia renders claims 1, 6, 11, and 16 unpatentable. In particular, Appellants argue that Bak's disclosure of assigning a new priority to a boosted thread at substantially any time and Gosalia's disclosure of a coprocessor does not teach enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU. (App. Br. 6-9; Reply Br. 2-4.)

3. Appellants contend that the Examiner erred in concluding that the combination of Bak and Gosalia renders claims 2, 7, 12, and 17 unpatentable. In particular, Appellants argue that Bak's disclosure of boosting a second thread execution to match a first thread execution priority does not teach wherein when the priority boosting of the first thread is enhanced, the first thread becomes the next thread to be dispatched for execution. (App. Br. 9-10.)

4. Appellants contend that there is no motivation for the proffered combination. (Reply Br. 4-5.)

Examiner's Findings and Conclusions

1. The Examiner concludes that the combination of Bak and Gosalia renders the claimed invention unpatentable. In particular, the Examiner finds that the critical reference date of Gosalia is the filing date of U.S. Provisional Patent Application 60/448,402. (Ans. 8-9.)

2. The Examiner concludes that the combination of Bak and Gosalia renders claims 1, 6, 11, and 16 unpatentable. In particular, the

Examiner alleges that there is no support in Appellants' original disclosure for a first thread scheduled to run on a first CPU and rescheduling the first thread to run on a second CPU. (Ans. 9.) Further, the Examiner finds that Bak's disclosure of boosting priority of a first thread, in conjunction with its disclosure of a runtime environment generally executed using processors, teaches enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU. (Ans. 9-10.)

3. The Examiner concludes that the combination of Bak and Gosalia renders claims 2, 7, 12, and 17 unpatentable. In particular, the Examiner finds that Bak's disclosure of threads with higher execution priority that obtain the right to study an object before threads with a lower execution priority teaches wherein when the priority boosting of the first thread is enhanced, the first thread becomes the next thread to be dispatched for execution. (Ans. 10.)

II. ISSUE

The pivotal issue before us is therefore whether Appellants have shown that the Examiner erred in concluding that the combination of Bak and Gosalia renders the claimed invention unpatentable. In particular, the issue turns on whether the proffered combination teaches enhancing the priority boosting of the first thread by rescheduling the first thread to run on the second CPU, as recited in independent claim 1.

III. FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence:

Appellants' Invention

1. The present invention provides an algorithm that may be used to enhance priority boosting of a scheduled thread. (Spec. 7, ll. 20-22.) According to the invention, after passing priority to TH_1, TH_2 may also hand off its CPU time to TH_1 by pulling TH_1 into CPU_1's run queue before it goes to sleep. (Spec. 7, ll. 23-25.) Based on this priority, TH_1 may be dispatched for execution right after it has been pulled into CPU_1's run queue if it has already been awakened. (Spec. 7, ll. 25-28.) Thus TH_5, the next thread scheduled to run on CPU_1, will now have to wait before being dispatched for execution. (Spec. 7, ll. 29-30.)

Bak

2. As depicted in the flowchart of Figure 11, the operating system with which the obtained boosted thread is associated may assign a new priority to the obtained boosted thread at substantially any time (992). (Col. 23, ll. 24-27.) By way of example, if a first thread with an original priority of "2" is boosted to a priority of "6" by a second thread, and is then assigned a priority of "4" by the operating system, when the second thread unboosts the first thread, the first thread is unboosted to a priority of "4". (Col. 23, 27-31.)

3. As depicted in Figure 12, runtime environment (1135) may generally be executed using a processor or processors such as CPUs (1032). (Col. 24. ll. 55-57.)

Gosalia

4. Gosalia discloses systems and methods for scheduling the processing of a coprocessor whereby applications can submit tasks to a scheduler, and the scheduler can determine how much processing each

application is entitled to as well as an order for processing. (Abstract.) The invention also provides various techniques for determining whether a particular task is ready for processing. (Abstract.)

IV. PRINCIPLES OF LAW

Obviousness

"On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence of secondary indicia of nonobviousness." *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998).

> Section 103 forbids issuance of a patent when "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

KSR Int'l Co. v. Teleflex Inc., 550 U.S. 398, 406 (2007).

In *KSR*, the Supreme Court emphasized "the need for caution in granting a patent based on the combination of elements found in the prior art," and discussed circumstances in which a patent might be determined to be obvious. *Id.* at 415 (citing *Graham v. John Deere Co.*, 383 U.S. 1, 12 (1966)). The Court reaffirmed principles based on its precedent that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Id.* at 416. The operative question in this "functional approach" is thus "whether the improvement is more than the predictable use of prior art elements according to their established functions." *Id.* at 417.

In identifying a reason that would have prompted a person of ordinary skill in the relevant field to combine the prior art teachings, the Examiner must show some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *KSR,* 550 U.S. at 418 (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)) .

V. ANALYSIS

Claim 1

Independent claim 1 recites, in relevant part, enhancing the priority boosting of the first thread to run on the second CPU.

As set forth in the Findings of Fact section, Bak discloses that a first thread with an original priority of “2” may be boosted to a priority of “6” by a second thread. (FF 2.) Further, Bak discloses that the runtime environment (1135) may generally be executed using processors such as CPUs (1032). (FF 3.) We find that Bak’s disclosure teaches boosting priority utilizing multiple processors whereby a second thread boosts the priority of a first thread by passing the first thread its higher priority. However, the cited disclosure falls short of teaching or suggesting rescheduling the first thread to run on a second CPU. As noted above, this step ensures that a first thread receives processing time right away (FF 1.) We find that such an endeavor is not contemplated by the Bak reference. Further, we find that Gosalia’s disclosure does not cure this deficiency. While Gosalia discloses that applications can submit tasks to a scheduler and the scheduler can determine the order and entitlement of each process (FF 4), it is, however, silent on rescheduling the first process to run on a second CPU. Since Appellants have shown at least one error in the rejection of claim 1, we need not reach the merits of Appellants' other arguments. Therefore, it follows that Appellants have shown that the Examiner erred in concluding that the combination of Bak and Gosalia renders independent claim 1 unpatentable.

Because claims 2 through 20 also incorporate the limitation discussed above, we find that Appellants have also shown error in the Examiner's rejection of these claims for the reasons set forth in our discussion of independent claim 1.

VI. CONCLUSION OF LAW

Appellants have shown that the Examiner erred in rejecting claims 1 through 20 as being unpatentable under 35 U.S.C. § 103(a).

VII. DECISION

We reverse the Examiner's decision to reject claims 1 through 20 as being unpatentable under 35 U.S.C. § 103(a).

REVERSED

PEB

IBM CORPORATION- AUSTIN (JVL)
C/O VAN LEEUWEN & VAN LEEUWEN
PO BOX 90609
AUSTIN, TX 78709-0609